DTU Library

Technical Report on Autonomous Mobile Robot navigation

Özkil, Ali Gürcan

Publication date:
2009

Technical Report on
Autonomous Mobile Robot Navigation

Ali Gürcan Özkil

Marts 2009

Table of Contents

1 Objective

This report is the outcome of the project 'Nestor', which, in general terms, aims to utilize an autonomous mobile robot navigation system <u>for real world settings</u>. The report has three aims:

1. To outline the problem setting.
2. To layout the related concepts
3. To give the state of the art for dealing with these concepts.

2 Introduction

Before getting into the details about navigation, it is important to characterize the genera robot control problem. The following section describes the basics of this issue

3 Robotic Paradigms

A paradigm *is a philosophy or set of assumptions or techniques which characterize an approach to a class of problems* [1]. In this sense, the aim of a robotic paradigm is to organize the 'intelligence' of the system and control its actions.

A robotic system has three main set functions: SENSE, PLAN and ACT. SENSE functions gather information from robot's sensors and produce a useful output for other functionalities. PLAN functions take these sorts of outputs or use robot's own knowledge to produce a set of tasks for the robot to perform. ACT functions produce actuator commands to carry out physical embodiment with the environment.

There are currently three paradigms in robot control, which are described by the relationship between these three primitive functionalities (Figure 1).

3.1 Hierarchical paradigm

Also called as the classical/traditional artificial intelligence paradigm, it is historically the oldest method of organizing intelligence in mainstream robotics [2]. **Error! Reference source not found.**Since the very first implementation [3], it has been a dominating way of controlling robots through a logical sequence of actions.

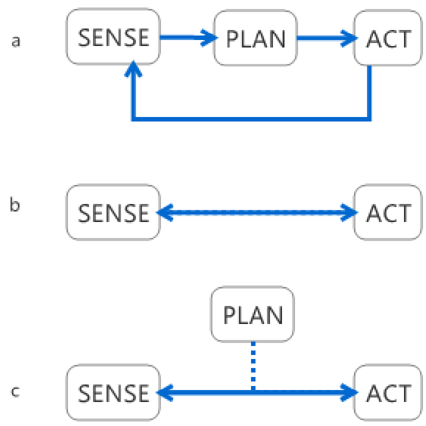

Figure 1, three robotic paradigms; (a) hierarchical, (b) reactive, (c) hybrid

Under this paradigm, the robot basically senses the world, plans its action, and then acts. Therefore, at each step it explicitly plans the next move. This model tends to construct a database to gather a global world model based on the data flow from the sensors, such that the planner can use this single representation to route the tasks to actions.

3.2 Reactive paradigm

Reactive paradigm came out as a reaction to the hierarchical paradigm in 80s. Hierarchical approach was based on an introspective view of how people think in a top-down manner. Reactive approach, on the other hand, utilized the findings of biology and cognitive physiology; which examined the living examples of intelligence [2].

In this approach, sensing is directly coupled to actuation, and planning does not take place. There are multiple instances of SENSE-ACT couplings, which can be also called as behaviors. The resulting action of the robot is the combination of its behaviors.

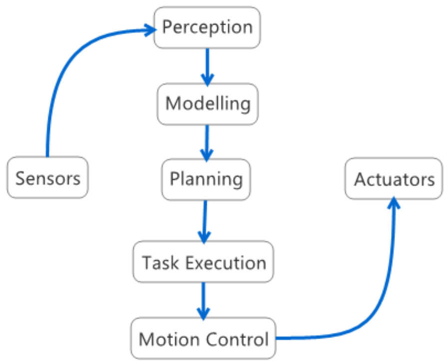

Figure 2, hierarchical paradigm in detail

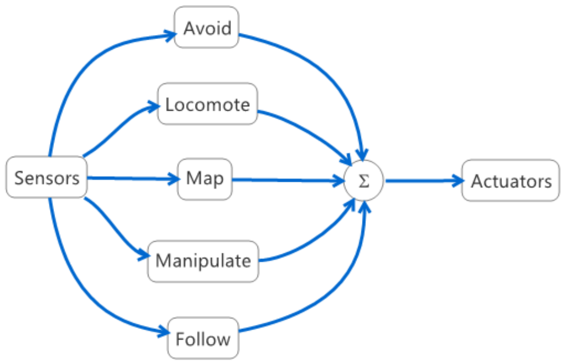

Figure 3, a reactive control paradigm example

Brooks, in his seminal paper [4], described the main difference between these two approaches as the way they decompose the tasks. According to him, reactive systems decompose tasks in layers. They start with generating basic survival behaviors and then evolve new ones that either use the existing ones or create parallel tracks of more advanced ones. If anything happens to the advanced ones, the lower behavior will still operate, ensuring the survival of the system. This is similar to the functionalities of human brain stem such as breathing, which continue independently from high level cognitive functions of the brain (i.e. talking), or even in case of cognitive hibernation (i.e. sleeping)

Purely reactive systems showed the potential of the approach, but it was seen that it is not very suitable for general purpose applications without any planning.

3.3 Hybrid Paradigm

Hybrid approach was first exemplified by Arkin in 90s to address the shortcomings of the reactive approach [5]. In this approach planning occurs concurrently with the sense-act couplings in such a way that tasks are decomposed to subtasks and behaviors are accordingly generated. Sensory information is routed to requesting behaviors, but it is also available to the planner for building a task oriented world model. Therefore, sensing is organized as a mixture of hierarchical and reactive styles; where planning is done at one step and sensing and acting are done together.

The hybridization brought up several architectural challenges, such as how to distinguish reaction and deliberation, how to organize deliberation, or how the overall behavior will emerge. Several architectures have been developed to tackle these issues, most of which mainly focused on behavioral management. It was found out that two primary ways of combining behaviors; subsumtion [4] and potential field summation [6] are rather limited, so other methods based on voting (DAMN) [7] , fuzzy logic (Saphira) [8] and filtering (SFX) [9] were introduced. The book 'Behavior Based Robotics [10]' is regarded as the most complete work on AI robotics, with a comprehensive list of such robot architectures explored in detail [2].

4 Autonomous Navigation

Autonomous mobile robot navigation can be characterized by three questions [11]:

- Where am I?
- Where am I going?
- How do I get there?

In order to tackle these questions, the robot has to:

- handle a map of its environment
- Self localize itself in the environment
- Plan a path from its location to a desired location

Therefore the robot has to have a model of the environment, be able to perceive, estimate its relative state and finally plan and execute its movement.

An autonomous robot navigation system has traditionally been hierarchical, and it consists of a dynamical control loop with four main elements: Perception, Mapping/localization, Cognition and Motor Control (Figure 4).

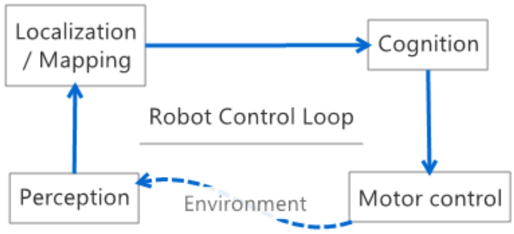

Figure 4, autonomous navigation problem

This chapter aims to summarize these elements and give an overview of relevant problems to be addressed.

4.1 Perception

First action in the control loop is perception of the self and the environment, which is done through sensors. Proprioceptive sensors capture information about the self-state of the robot, whereas exoprioceptive sensors capture information about the environment. Types of sensors being used on mobile robots shows a big variety [12,13]. The most relevant ones can be briefly listed as: encoders, gyroscopes, accelerometers, sonars, laser range finders, beacon based sensors and vision sensors.

In theory, navigation can be realized using only proprioceptive sensors, using odometry. It is basically calculating the robot position based on the rotation of wheels and/or calculating orientations using gyroscopes/accelerometers. But in real world settings, odometry performs poorly over time due to unbounded growth of integration errors caused by uncertainties.

It is also possible to navigate using only exoprioceptive sensors. One such realization of this approach is the Global Positioning System (GPS); which is being successfully used in vehicle navigation systems. The problem with GPS and its upcoming, European counterpart Galileo [14] is that these systems require a direct line of sight to the satellites on earth orbit. Therefore these systems are especially inapplicable to indoor applications.

Shortcomings of GPS system led researchers to several ground based approaches. Several alternatives have been developed based on i.e.: Radio beacons[15], Wireless Ethernet[16], GSM networks [17], Wireless Sensor Networks (WSN)[18] , RFID tags [19], barcodes [20] or laser reflectors [21]. Such methods can ease the problem of navigation, but they might require substantial amount of environmental modification.

This makes them inflexible and costly to install and maintain. Due to such reasons, many researches focused on solving the robot navigation problem in unmodified environments.

Many of the state of the art techniques for navigation in unmodified environments uses combinations of proprioceptive and exoprioceptive sensors and fuse them using probabilistic techniques. Sonars, laser range finders and several kinds of vision sensors are used to capture information in such methods.

4.1.1 Environmental Representation

How the environment is represented is an important factor in navigation. It depends on several characteristics of the sensor and the data acquisition system, such as range and resolution, update speed, bandwidth. It also characterizes how it is stored in case of mapping or map handling.

Simplest way of representing an environment is using raw sensor data. Information coming from sensors i.e. laser range scans, are sequentially stored in the same data type they are acquired. In such a generalized case, the problem is the low distinctness of the data. Also, it eventually results in a large volume of data with time, which brings up computational challenges.

Alternatively, features can be used for modeling. Complexity level of features is an important factor for navigational purposes. Using low level features such as lines/circles will generate a smaller database compared to the previous approach, yet with a moderate amount of ambiguity associated. More complex features in the forms of i.e. patterns/objects can even decrease the size of the database with lesser ambiguities. But too much complexity, on the other hand, can have two adverse effects. First, it might yield difficulties in detection and require high computational resources. And secondly, it will result in very small databases, which might not entirely capture the characteristics of the environment.

Range sensors have been the dominating choice for environmental sensing on robots. Early works extensively used sonar arrays for distance sensing, but the limitations with range and resolution of sonars severely affected functions of mapping and localization. Time-of-flight laser scanners later became widely applicable to mobile robotics, but their scanning field is restricted to a horizontal plane, which in turn yields to poor world representation [22]. This limitation was tackled by using oscillating the laser scanners [23][24][25] or multiple lasers with complementary placements[26] to achieve higher dimensionality in range sensing . Yet, these systems are rather expensive and complex to utilize in a real world robotic application. Finally, different vision based approaches has been emerged in the last decade to extract metric information from the environment using imaging sensors. Stereo systems have been long investigated for 3D range sensing, whereas a big amount of recent work is based on monocular systems that can extract metric information from the optical flow detected by the camera.

4.1.2 Maps Used in Mobile Robot Navigation

Idea of using maps for mobile robot navigation has been existed for quite some time, and roboticists have developed several types of maps for different needs based on how they can represent the environment. Buschka [27] classifies existing map types as follows:

> *Metric Maps:* Maps that carry distance information that corresponds to actual distances in the environment. Such a map can give a distance of a path or size of an object.
>
> *Topological Maps*: Maps where the environment is modeled according to its structure and connectivity, and often represented as a connectivity graph.
>
> *Sensor Level Maps:* Maps that are derived directly from the interpretation of the sensor inputs from the current position. (i.e. [28])
>
> *Appearance Based Maps:* Maps that functionally describe a position from sensor data. (i.e. [29])
>
> *Semantic Maps:* Maps which are oriented for high level decision making, and contain information about objects and their relationships with the environment. (i.e. [30-32])
>
> *Hybrid Maps:* A combination of different types of maps. Hybrid maps also need to glue elements that represent the same part of the environment in combined maps.

Following section elaborate on metric, topological and hybrid maps, with are the most commonly used types in mobile robotics.

4.1.2.1 Metric Maps:

From the control perspective, metric maps are useful when metric accuracy is necessary for i.e. precise localization or optimal path planning. Depending on the environmental representation, a metric map can be either feature based or grid based [33].

A metric feature based map is basically built upon the features that can be reliably observed in the environment. In [34-36]typical features of indoor environments such as walls, edges or corners are used for mapping of indoor environments.

In a grid based metric map, environment is divided in to a matrix of sub cells, where each cell represents a portion of the environment. A cell is considered to be occupied if an object exists in the corresponding area in the environment. Moravec and Elfes developed a common way of representing occupancy is using probabilities [37]. Saffiotti used fuzzy sets where occupancy and emptiness values are held separately [38]. In [39], grid is represented using histograms where each cell holds a value of how often a sensor has detected it. Stachniss and Burgard developed a coverage map, where each cell holds a value representing how much it is covered by an obstacle [40].

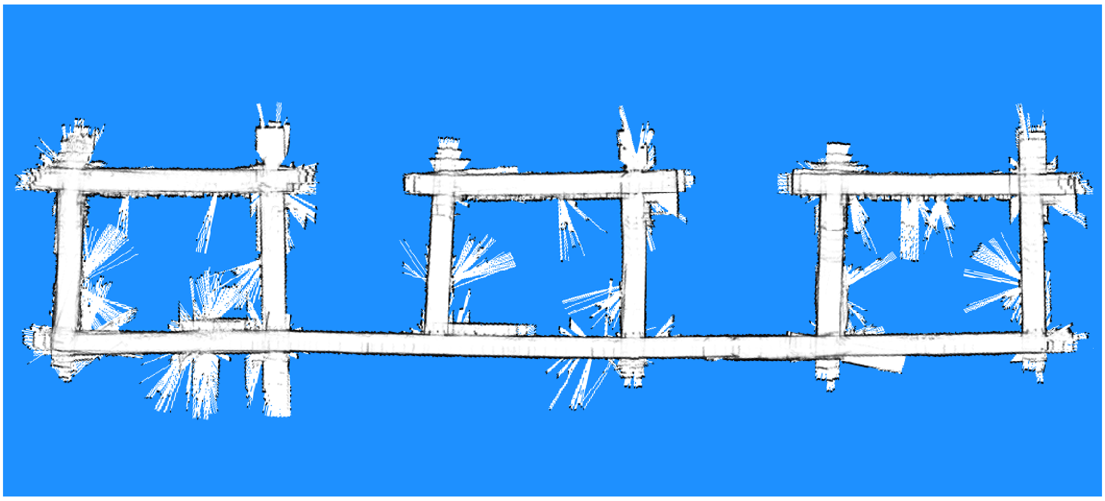

Figure 5, metric grid map of DTU 402-404

4.1.2.2 Topological Maps:

Topological maps describe how places in the world are connected or related to each other, thus represents the structure of the environment. Two elements that constitute a map are nodes; which represent the places, and edges; which represent connectivity. In practice, most topological maps are also augmented with some metric information on its nodes. Due to its simplicity in construction, topological maps are better suited for problems that require searching. (See Figure 6 for an example)

Apart from robots, humans might also need to interact with the topological maps for robot navigation. Different characteristics of the environment have been used by researches such as rooms or corridors as nodes and doors or passageways as edges. Thrun [41] preferred to use places with ‘significant features’ as nodes. Fabrizi [42] defined a node as a ‘large open space’. Duckett [43] proposed a system where a new node is placed after robot has travelled far away from the previous one.

Topological maps, such as reactive control paradigm, were inspired by biological studies of insects and animals. It can be also claimed that a topological map will be the best suitable for a behavior based navigation system.

Figure 6, a topological map: S-tog network in Copenhagen

4.1.2.3 Hybrid maps:

Since metric and topological maps are of fundamentally different types, both have advantages over each other. Table 1 summarizes this comparison.

It is clear to see that what is an advantage for one approach is a disadvantage to the other, which constituted the motivation to develop hybrid maps. The idea came to the scene as early as 1978 [44], but it has only been a decade that such maps emerged in an increasing number.

Table 1, Comparison of Metric and Topological maps [27]

	Metric Maps	Topological Maps
Pros	• High accuracy for localization and path planning • Possibility to make optimal routes • Easy to build, represent and maintain for small environments • Layout is easily readable for humans	• Easy to scale up for large environments • Very suitable for planning • Sensor precision and reliability is not as important • No need for precise position estimation for map building • Good interface to symbolic problem solvers
Cons	• Difficult to scale up for large environments • Costly path planning • Need for reliable sensors • Need for precise position estimate for map building.	• Low accuracy • Possible suboptimal paths • Difficult to build and maintain

Two types of hybrid maps are parallel maps and patchwork maps. A parallel map constitutes of at least two different maps that represent the same area in an environment. Most parallel maps are constructed automatically by extracting a topological map from a metric one. Thrun utilized Voronoi diagrams in the empty parts of a grid map in [41]. A similar approach was carried in [42] by using image processing. The opposite approach, extracting metric maps from topological nodes had also been presented [45]. An interesting multi-layered hierarchical parallel map representation is developed in [46,47] where the main focus was efficient localization. The map is called 'Annotated Hierarchical graph' and it consists of hierarchically ordered topological maps, supported with local metric patches in the lowest layer. Nieto also developed a novel kind of parallel map, which consisted of an augmentation of a feature based metric map and a grid based metric map [48]. While the first were used for localization, the latter was used for optimal route planning.

A patchwork map is a representation, where the environment is globally symbolized by a topological map and a set of metric map patches. This kind of maps can be easily scaled up, thus representing really large environments; yet perform fine metric localization due to the patches.

Several patchwork maps simply connect small sized metric maps based on topology [49]. Thus nodes do not correspond to any particular environmental structure. More elaborate patchworks used openings between i.e. rooms and corridors as the node features [50]. In [51,52] similar approach is used for node selection, and the rest of the topology is completed using 'Reduced Generalized Voronoi Graph'. Aguirre developed a complex patchwork map in [53], where two kinds of metric maps where used in each room, which acted as nodes in the topology.

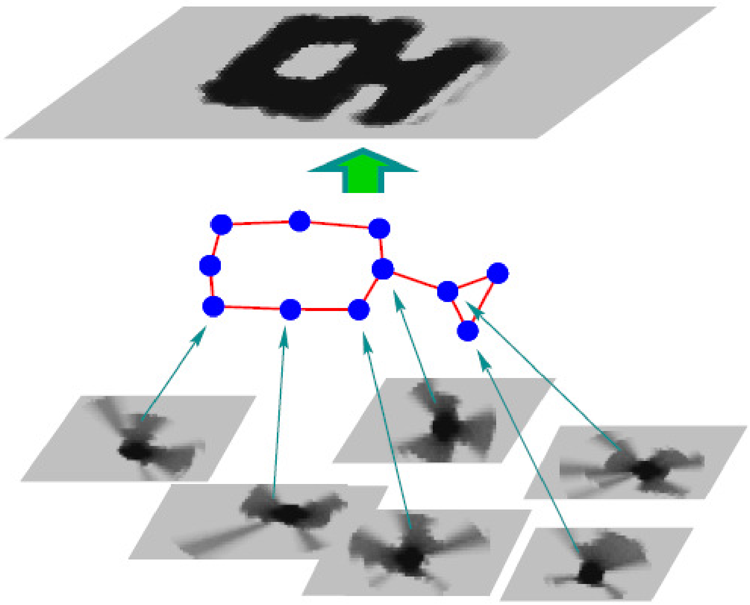

Figure 7, a hybrid map: Global metric map is extracted from the 'signatures' of topological nodes [45]

4.2 Mapping and Localization

4.2.1 Mapping

A mobile robot requires a representation of the environment for autonomous navigation in the form of a map. Based on the environment characteristics and the type of the map, it is possible to build robot maps using existing maps by other means. But in most of the cases, the robot needs to build a map of the environment in a subsequent training phase.

Metric grid maps are the most commonly used types of maps in mobile robot navigation. Building metric maps basically requires estimating the initial position of the robot, and updating the cells of the map as the new sensory information is acquired. The most trivial approach is to use odometry for position estimation. As explained previously, estimation error accumulates by time in odometrical systems. The apparent idea [54] to address this problem is to use the map, which is being built at that moment, for correcting the estimation, which is now coined as *Simultaneous Localization and Mapping* – SLAM.

Particularly difficult part of the SLAM problem with the grid maps is that the cell positions in a grid map are static. Therefore, if a robot recognizes a place it has already been during mapping (loop closure), it might see that its position is off and needs to be corrected. On the other hand, to correct the grid map, the entire map should be traced back and recalculated based on the new information.

An evident method is to build the map sequentially by first localizing and then rebuilding the map based on adjusted positions [55]. Genetic algorithms are also used for mapping. Duckett developed a method [56] where several maps are generated with slightly altering paths, and then a genetic algorithm is used to select the best maps and combine new paths to test. In [57], a new grid map representation is generated where the cells are able to hold multiple hypotheses about the map. The least probable hypotheses are later removed in a map update stage. Rao-Blackwellized particle filters, introduced in [58] became a popular choice for building grid maps. In this approach, a number of maps based on single particles are being carried and updated simultaneously. Recent improvements on this method permitted to reduce the number of particles to still get good results [59-61].

Feature maps differ from grid maps in the sense that sensor data is used to extract features before the mapping stage. These features are then compared to the ones in the map so that either the new feature is added to the map or the existing features in the map are updated accordingly, or used for correcting the position estimation. Many of the solutions are based on the approach presented in [62]. The most significant developments around this method are based on how the Kalman filter is utilized for position update. Information filter is introduced to ease the computation burden in [63,64]. Also unscented filter is used in [65,66] to cope with the nonlinearities.

Building topological maps can be done in two different ways; by using sensor data or by using another type of map. Choset used a generalized Voronoi graph as a map in [51]. The map is constructed my moving the robot in the environment to construct the nodes of the map, and visited nodes are detected by matching their “signatures” to the previously acquired ones. Thrun et.al used the latter technique in [52], where they preprocessed the grip maps to threshold the occupancy probabilities to further generate Voronoi diagrams on the empty areas of the map. Local minima found in the Voronoi graph are used to partition the grid map into nodes.

4.2.2 Localization

Localization is the task of finding the position of a mobile robot in an environment, based on its representation.

4.2.2.1 Metric localization

Localization can be defined as the task of estimating the robot's pose in the world, given a-priori map. The estimate, also referred as *belief*, is often augmented with some measure of uncertainty that can arise from several factors. The belief is updated when the robot performs an action or makes an observation. A robot action (i.e. movement) increases the uncertainty (due to integration errors), whereas observations often reduce the uncertainty of the robot pose.

Localization is tightly coupled to how belief is represented and estimated. Most of the robotic systems use planar maps. The main reason for that is to decrease the complexity of the problem by reducing dimensionality of the robot pose vector from 6-D (x, y, z, pitch, roll, yaw) to 3-D (x, y, yaw).

One approach to solve the problem is position tracking, where the belief of the robot is reduced to a single pose. The position is estimated in a single hypothesis, and whenever an action or observation occurs, the hypothesis is updated. Therefore, the initial position of the robot must be known to be able to track the position. Kalman filter [67], and its variants are widely used in position tracking. In [68] sonars range finders are used for line extraction and a Kalman filter is used for matching. In [69], an extended Kalman Filter is used to match raw sensor data with a feature based metric map. Fuzzy logic is also used for representing uncertainty in position tracking in [70].

Position tracking problem deals with a single pose, therefore representation is simple, and update calculations are computationally cheap. But this technique requires that the initial position is known. In addition, if the measurements become vague, the position can be lost.

The alternative solution to position tracking is to represent multiple hypotheses of the pose, which is often called as global position estimation. In this approach, the initial position is not needed to be known, but due to the high degrees of uncertainities imposed by multiple hypotheses about the pose of the robot, position estimation is computationally expensive.

In [71] several position candidates are tracked using Kalman filters. The number of hypotheses adapts the uncertainty of the localization. Safiotti et. al. used fuzzy sets to represent uncertainities to carry out multi-hypothesis tracking in [72]. In [73] Markov localization is introduced, where each cell of a grid map holds a belief of how much the actual position of the robot is in that cell. In this approach, the localization grid map represents a probably density function (pdf) of the belief of localization. As the robot moves or observes, cells are updated using Bayesian updating. In [74] , the method is further modified to overcome the heavy computational cost of the approach. An alternative is proposed in [75], where the updating is based on fuzzy logic instead of Bayesian inference. In [76] pdf used in localization belief is represented as a set of samples. This approach reduced the number of calculations compared to Markov localization, while it is still possible to perform global localization. This approach is called Monte Carlo localization, and further improved in [77-79] to decrease the number of samples needed.

4.2.2.2 Topological Localization

A topological map is consisted of nodes and edges. Therefore, topological localization is the task of finding in which edge or node the robot is. Apart from the environmental representation (i.e. how the edged and nodes are defined), topological localization requires reliable place recognition and detection of edge traversal. In [80], nodes are defined based on the sudden changes in the behavior pattern of the robot. For instance, if the robot is following a wall, and after a while it encounters an obstacle so it as to perform another action, that particular place is defined as nodes. Nodes are identified using features such as distance travelled since the last node and the 'signature' of the node given by sonar sensors. In [81], the nodes are recognized using a similar signature approach, and then these signature are learned using a growing neural network. Localization is then performed using signature matching and odometry. In [82], a local topological map is built and then compared to a global one, to obtain the most likely position. In [83], an omnidirectional camera is used for performing topological localization. Queried images are compared to images stored in the map. Image histograms are used as global features for image representation, thus the amount of information stored is highly reduced.

4.3 Cognition and Path planning

Robot cognition very much depends on the general use of the robot. In the context of this report, problem setting is defined as autonomous mobile robot navigation. Therefore, the discussion is focused on path planning for navigation.

Path planning can be defined as searching a suitable path in a map from one place to another. Depending on the map type, it is possible to follow different strategies for planning paths.

Metric maps are useful for planning precise paths. Due to metric information associated, it is possible to find nearly-optimal paths using metric maps [27]. There exists several different methods for path planning, but they are based on a few general approaches. Latombe classifies these approaches in [84]as follows:

> *Road map*: a road map is a collision free set of path between a starting position and an ending position. Therefore, they describe the connectivity of robot free space on the map. One method to construct road map is based on visibility graphs [85]. In this method, path is incremented from one point to other points that are visible from the first point. Another method is to construct a Voronoi graph, which tries to maximize the clearances between the robot and obstacles [86].
>
> *Cell decomposition*: free space in the map is divided into non-overlapping cells, and a connectivity graph describes how the cells are connected to each other. The result is a chain of cells, which also describes the path. Therefore, formation of cells plays an important role in planning the path. In [84], trapezoidal decomposition is used, where a polygonal map is divided into trapezoidal cells by generating vertical line segments at each corner of every polygon. In [87] qualitative spatial reasoning is used for path planning, which is inspired of the way humans find their paths with imprecise knowledge. Cell decomposition is also a suitable method for area coverage, where the planner breaks down the target area into cells to be all traversed. Applications of this approach can be listed as i.e. lawn moving, snow removal or floor cleaning [88]
>
> *Potential field*: A potential field function is defined an applied over the free space on the map, where the goal acts as an attractive potential (sink) and the obstacles act as repulsive potentials (sources). The path is then derived based on the derivative of the potential field, where the steepest direction is followed. This approach was first developed for online collision avoidance in [89]. It is combined with a graph search technique in [84] for path planning.

Topological maps are well suited for planning paths. Graph search algorithms, such as A* [90] or D*[91], can be used the plan the shortest path on a topological map. In most of the cases, the number of edges and nodes are moderate, so the path planning can be performed very quickly. Path finding time is even further shortened in [92] by preprocessing all paths and storing them in a lookup table. In [82] wave-front algorithm is used for both path planning and collision avoidance. In [93,94], planning on very large maps is described in the context of hierarchical topological maps.

4.4 Motion Control

Motion control is the final phase in the robot control loop, where the high level plans generated in the previous phase are translated into robot movements. Therefore, this level processes abstract motion commands and produces low level commands for controlling motor speeds.

Obstacle avoidance is of particular interest, and it can be classified under motion control. It is one of the key issues to successful mobile robot applications, as it ensures the safety of both robot and surrounding entities. Obstacle avoidance strategies range from primitive algorithms that just stop the robot when an obstacle is detected; to complex ones that enable robot detour the obstacles.

Borenstein introduced vector field histogram (vfh) algorithms fro obstacle avoidance tasks in [39], based on local potential fields. In this approach, first the range data is continuously sampled, and a two dimensional local grid is generated to represent the environment. In the next stage, one dimensional polar histogram is extracted from the local grid in terms of angular sectors with particular widths. Finally, this one dimensional histogram is threshold and the angular sector with the highest density is selected as the direction. Sped of the robot is also adjusted in correlation with the distance from the obstacle. In [95], the algorithm is improved by incorporating the kinematics of the robot as the original algorithm assumes that the robot is able to change its direction instantaneously (named as vfh+). The algorithm is further improved and coined as vfh* in [96]. In contrast to vfh and vfh+, which are purely local algorithms based on current sensor readings, vfh* incorporated A* graph search algorithm to consider more than immediate surroundings of the robot.

In [97], dynamic window approach is introduced as an obstacle avoidance method. Kinematic constraints of a Synchro drive robot are taken into account by directly searching the velocity space of the robot. The search space is further reduced to a dynamic window, which contains those velocities that can be achieved by the robot, given its velocity and acceleration. Finally, this window is searched for a velocity, which aligns with the target direction of the robot. In [98], the method is adapted to holonomic robots, which allowed high speed obstacle avoidance with high maneuverability.

Finally, nearness diagram is introduced in [99], which is based on heuristic rules that are inferred from possible high and low safety situations that the robot can end up. Based on five rules (two low and three high safety situations), five behaviors are defined, where robot compares its current situation to these predefined ones and executes the appropriate behavior. It is shown in [100] that this reactive approach can perform well in cluttered environments with narrow passages, as compared to previous approaches.

5 Bibliography

[1] "paradigm - Definition from the Merriam-Webster Online Dictionary."

[1] R. Murphy, *Introduction to AI Robotics*, MIT Press, 2000.

[3] N.J. Nilsson, *Shakey the Robot*, Storming Media, 1984.

[4] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, 1986, pp. 14-23.

[5] R. Arkin and T. Balch, "AuRA: principles and practice in review," Apr. 1997.

[6] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobilerobot navigation," *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 1991, pp. 1398-1404.

[7] J.K. Rosenblatt, "DAMN: a distributed architecture for mobile navigation," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, 1997, pp. 339-360.

[8] K. Konolige and K. Myers, "The saphira architecture for autonomous mobile robots," 1998.

[9] R.R. Murphy and R.C. Arkin, "Sfx: An Architecture For Action-oriented Sensor Fusion," *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, 1992.

[10] R.C. Arkin, *Behavior-Based Robotics*, MIT Press, 1998.

[11] J. Borenstein, H.R. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques*, AK Peters, Ltd. Natick, MA, USA, 1996.

[12] R. Siegwart and I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, MIT Press, 2004.

[13] H.R. Everett, *Sensors for mobile robots: theory and application*, AK Peters, Ltd. Natick, MA, USA, 1995.

[14] "Energy and Transport - GALILEO."

[15] J.C. Harris, H.T. Inc, and V.A. Clifton, "An infogeometric approach to telerobotics," *Telesystems Conference, 1994. Conference Proceedings., 1994 IEEE National*, 1994, pp. 153-156.

[16] A.M. Ladd, K.E. Bekris, A. Rudys, L.E. Kavraki, and D.S. Wallach, "Robotics-Based Location Sensing Using Wireless Ethernet," *Wireless Networks*, vol. 11, 2005, pp. 189-204.

[17] K. Briechle and U.D. Hanebeck, “Localization of a mobile robot using relative bearing measurements,” *Robotics and Automation, IEEE Transactions on*, vol. 20, 2004, pp. 36-44.

[18] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, 2002, pp. 393-422.

[19] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, “Mapping and localization with RFID technology,” *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*.

[20] I. Ulrich and I. Nourbakhsh, “Appearance-based place recognition for topological localization,” *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, IEEE; 1999, 2000, pp. 1023-1029.

[21] A. Howard, M.J. Mataric, and G.S. Sukhatme, “Localization for mobile robot teams: A maximum likelihood approach,” *USA: University of Sourthern California*, 2001.

[22] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, “Omnidirectional Vision Based Topological Navigation,” *International Journal of Computer Vision*, vol. 74, 2007, pp. 219-236.

[23] H. Surmann, K. Lingemann, A. Nuchter, and J. Hertzberg, “A 3D laser range finder for autonomous mobile robots,” *Proceedings of the 32nd ISR (International Symposium on Robotics)*, 2001, pp. 153-158.

[24] H. Surmann, A. Nüchter, and J. Hertzberg, “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments,” *Robotics and Autonomous Systems*, vol. 45, 2003, pp. 181-198.

[25] C. Brenneke, O. Wulf, and B. Wagner, “Using 3D laser range data for SLAM in outdoor environments,” *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003.

[26] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications tomulti-robot and 3D mapping,” *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000.

[27] P. Buschka, *An Investigation of Hybrid Maps for Mobile Robots*, Örebro: Örebro universitetsbibliotek, 2005.

[28] A. Elfes, “A sonar-based mapping and navigation system,” *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, 1986.

[29] B.J.A. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura, “A probabilistic model for appearance-based robot localization,” *Image and Vision Computing*, vol. 19, 2001, pp. 381-391.

[30] R. Chatila and J. Laumond, “Position referencing and consistent world modeling for mobile robots,” *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985.

[31] G.L. Dudek, “Environment representation using multiple abstraction levels,” *Proceedings of the IEEE*, vol. 84, 1996, pp. 1684-1704.

[32] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernandez-Madrigal, and J. Gonzalez, “Multi-hierarchical semantic maps for mobile robotics,” *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 2278-2283.

[33] P. Jensfelt, *Approaches to Mobile Robot Localization in Indoor Environments*, Stockholm: KTH, Signals, Sensors and Systems, .

[34] J. Gasós and A. Martín, “A fuzzy approach to build sonar maps for mobile robots,” *Computers in Industry*, vol. 32, 1996, pp. 151-167.

[35] J.J. Leonard and H.F. Durrant-Whyte, “Simultaneous map building and localization for an autonomous mobilerobot,” *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, 1991, pp. 1442-1447.

[36] W.D. Rencken, “Autonomous sonar navigation in indoor, unknown and unstructuredenvironments,” *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, 1994.

[37] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985.

[38] A. Saffiotti, “The uses of fuzzy logic in autonomous robot navigation: a catalogue raisonne,” *Soft Computing*, vol. 1, 1997, pp. 180-197.

[39] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobilerobots,” *Robotics and Automation, IEEE Transactions on*, vol. 7, 1991, pp. 278-288.

[40] C. Stachniss and W. Burgard, “Exploring Unknown Environments with Mobile Robots using Coverage Maps,” *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, LAWRENCE ERLBAUM ASSOCIATES LTD, 2003, pp. 1127-1134.

[41] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, 1998, pp. 21-71.

[42] E. Fabrizi and A. Saffiotti, "Extracting topology-based maps from gridmaps," *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000.

[43] T. Duckett, S. Marsland, and J. Shapiro, "Fast, On-Line Learning of Globally Consistent Maps," *Autonomous Robots*, vol. 12, 2002, pp. 287-300.

[44] B. Kuipers, "Modeling spatial knowledge," *Cognitive Science*, vol. 2, 1978, pp. 129-153.

[45] T. Duckett and A. Saffiotti, "Building globally consistent gridmaps from topologies," *Proceedings of the Sixth International IFAC Symposium on Robot Control (SYROCO), Wien, Austria*, 2000.

[46] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernandez-Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 2278-2283.

[47] J.A. Fernandez-Madrigal, "Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment," *Integrated Computer-Aided Engineering*, vol. 11, 2004, pp. 309-322.

[48] J.I. Nieto, J.E. Guivant, and E.M. Nebot, "The HYbrid metric maps (HYMMs): a novel map representation for DenseSLAM," *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*.

[49] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*.

[50] H. Shatkay and L.P. Kaelbling, "Learning topological maps with weak local odometric information," *In Proceedings of IJCAI-97. IJCAI, Inc*, 1997.

[51] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): towardexact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, 2001, pp. 125-137.

[52] S. Thrun and A. Bücken, "Integrating grid-based and topological maps for mobile robot navigation," *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 1996, pp. 944-951.

[53] E. Aguirre and A. Gonzalez, “Integrating fuzzy topological maps and fuzzy geometric maps for behavior-based robots,” *International Journal of Intelligent Systems*, vol. 17, 2002, pp. 333-368.

[54] R. Chatila and J. Laumond, “Position referencing and consistent world modeling for mobile robots,” *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985.

[55] S. Thrun, W. Burgard, and D. Fox, “A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots,” *Autonomous Robots*, vol. 5, 1998, pp. 253-271.

[56] T. Duckett, “A Genetic Algorithm for Simultaneous Localization and Mapping,” *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, IEEE; 1999, 2003, pp. 434-439.

[57] A. Eliazar and R. Parr, “DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks,” *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, LAWRENCE ERLBAUM ASSOCIATES LTD, 2003, pp. 1135-1142.

[58] K. Murphy, “Bayesian map learning in dynamic environments,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, 1999, pp. 1015–1021.

[59] G. Grisettiyz, C. Stachniss, and W. Burgard, “Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling,” *Robotics and Automation, 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 2432-2437.

[60] G. Grisetti, C. Stachniss, and W. Burgard, “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters,” *IEEE TRANSACTIONS ON ROBOTICS*, vol. 23, 2007, p. 34.

[61] G. Grisetti, G.D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, “Fast and accurate SLAM with Rao–Blackwellized particle filters,” *Robotics and Autonomous Systems*, vol. 55, 2007, pp. 30-38.

[62] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” *Autonomous Robot Vehicles*, vol. 1, 1990, pp. 167-193.

[63] U. Frese and G. Hirzinger, “Simultaneous localization and mapping-a discussion,” *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2001, pp. 17–26.

[64] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng, *Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results*, Springer, 2002.

[65] K.S. Chong and L. Kleeman, "Mobile robot map building from an advanced sonar array and accurate odometry," *International Journal of Robotics Research*, vol. 18, 1999, pp. 20-36.

[66] R. Martinez-Cantin and J.A. Castellanos, "Unscented SLAM for Large-Scale Outdoor Environments," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 328–333.

[67] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, 1960, pp. 35–45.

[68] J.L. Crowley, "World modeling and position estimation for a mobile robot usingultrasonic ranging," *1989 IEEE International Conference on Robotics and Automation, 1989. Proceedings.*, 1989, pp. 674-680.

[69] J.J. Leonard and H.F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, 1991, pp. 376-382.

[70] J. Gasós and A. Saffiotti, "Using fuzzy sets to represent uncertain spatial knowledge in autonomous robots," *Spatial Cognition and Computation*, vol. 1, 1999, pp. 205-226.

[71] P. Jensfelt, "Localization using laser scanning and minimalistic environmental models," *Licentiate thesis, Automatic Control, Royal Institute of Technology, SE-100*, vol. 44, 1999.

[72] A. Saffiotti and L.P. Wesley, "Perception-based self-localization using fuzzy locations," *Lecture Notes in Computer Science*, 1996, pp. 368-384.

[73] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, 1998, pp. 195-208.

[74] W. Burgard, A. Derr, D. Fox, and A.B. Cremers, "Integrating global position estimation and position tracking formobile robots: the dynamic Markov localization approach," *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998. Proceedings.*, 1998.

[75] P. Buschka, A. Saffiotti, and Z. Wasik, "Fuzzy landmark-based localization for a legged robot," *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings*, 2000.

[76] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings*, 1999.

[77] D. Fox, "KLD-sampling: Adaptive particle filters and mobile robot localization," *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[78] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *To appear in Artificial Intelligence*, 2001.

[79] P. Jensfelt and T. högskolan i Stockholm, *Approaches to mobile robot localization in indoor environments*, Tekniska högsk., 2001.

[80] J. de Lope and D. Maravall, "Integration of reactive utilitarian navigation and topological modeling," 2003.

[81] U. Nehmzow and C. Owen, "Robot navigation in the real world: Experiments with Manchester's FortyTwo in unmodified, large environments," *Robotics and Autonomous systems*, vol. 33, 2000, pp. 223-242.

[82] C.H. Choi, J.B. Song, W. Chung, and M. Kim, "Topological map building based on thinning and its application to localization," *IEEE/RSJ International Conference on Intelligent Robots and System, 2002*, 2002.

[83] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00*, 2000.

[84] J.C. Latombe, *Robot motion planning*, Kluwer academic publishers, 1991.

[85] N.J. Nilsson, *A mobile automaton: An application of artificial intelligence techniques*, Storming Media, 1969.

[86] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, 2000, p. 96.

[87] C. Freksa, R. Moratz, and T. Barkowsky, "Schematic maps for robot navigation," *Lecture notes in computer science*, 2000, pp. 100-114.

[88] H. Choset, "Coverage for robotics – A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, Oct. 2001, pp. 113-126.

[89] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, 1986, p. 90.

[90] P.E. Hart, N.J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, 1968, pp. 100-107.

[91] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments."

[92] S. Thrun, J. Gutmann, D. Fox, W. Burgard, and B. Kuipers, "Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach," *AAAI/IAAI*, 1998, pp. 989-995.

[93] C. Galindo, J.A. Fernandez-Madrigal, and J. Gonzalez, "Improving efficiency in mobile robot task planning through world abstraction," *IEEE Transactions on Robotics*, vol. 20, 2004, pp. 677-690.

[94] J.A. Fernández-Madrigal, "Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment," *Integrated Computer-Aided Engineering*, vol. 11, 2004, pp. 309-322.

[95] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998.

[96] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with look-aheadverification," *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000.

[97] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, 1997, pp. 23-33.

[98] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 1999, pp. 341-346 vol.1.

[99] J. Minguez and L. Montano, "Nearness Diagram Navigation (ND): A New Real Time Collision Avoidance Approach," *In Proc. of the ieee/rsj international conference on intelligent robots and systems (iros'00*, vol. 60, 2000, pp. 2094--2100.

[100] J. Minguez and L. Montano, "Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 20, 2004, p. 45.